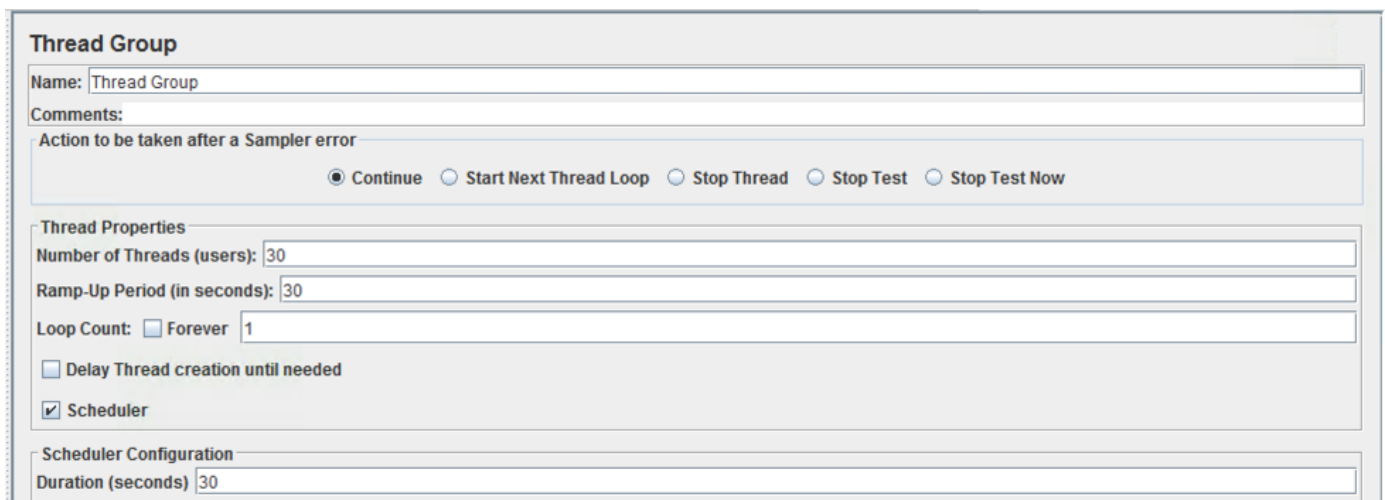


JMeter Test Results: Why the Actual Users Number is Lower than Expected

In [JMeter](#), we can't always tell the number of concurrent users we'll get for each [Thread Group](#) configuration.

For example, given the following configuration:

- Number of Threads: 30
- Ramp-up Period: 30
- Loop Count: 1
- Duration(seconds): 30



The screenshot shows the 'Thread Group' configuration window in JMeter. The 'Name' field is 'Thread Group'. The 'Comments' field is empty. The 'Action to be taken after a Sampler error' section has five radio buttons: 'Continue' (selected), 'Start Next Thread Loop', 'Stop Thread', 'Stop Test', and 'Stop Test Now'. The 'Thread Properties' section includes: 'Number of Threads (users):' set to 30, 'Ramp-Up Period (in seconds):' set to 30, 'Loop Count:' with 'Forever' unchecked and '1' entered, 'Delay Thread creation until needed' unchecked, and 'Scheduler' checked. The 'Scheduler Configuration' section has 'Duration (seconds):' set to 30.

How many concurrent users do you think you'll get?

Surprisingly (or not), there is no "right" answer. There are a

lot of criteria that impact the real concurrency, e.g. Connect time, Latency, and - above all - Response Time.

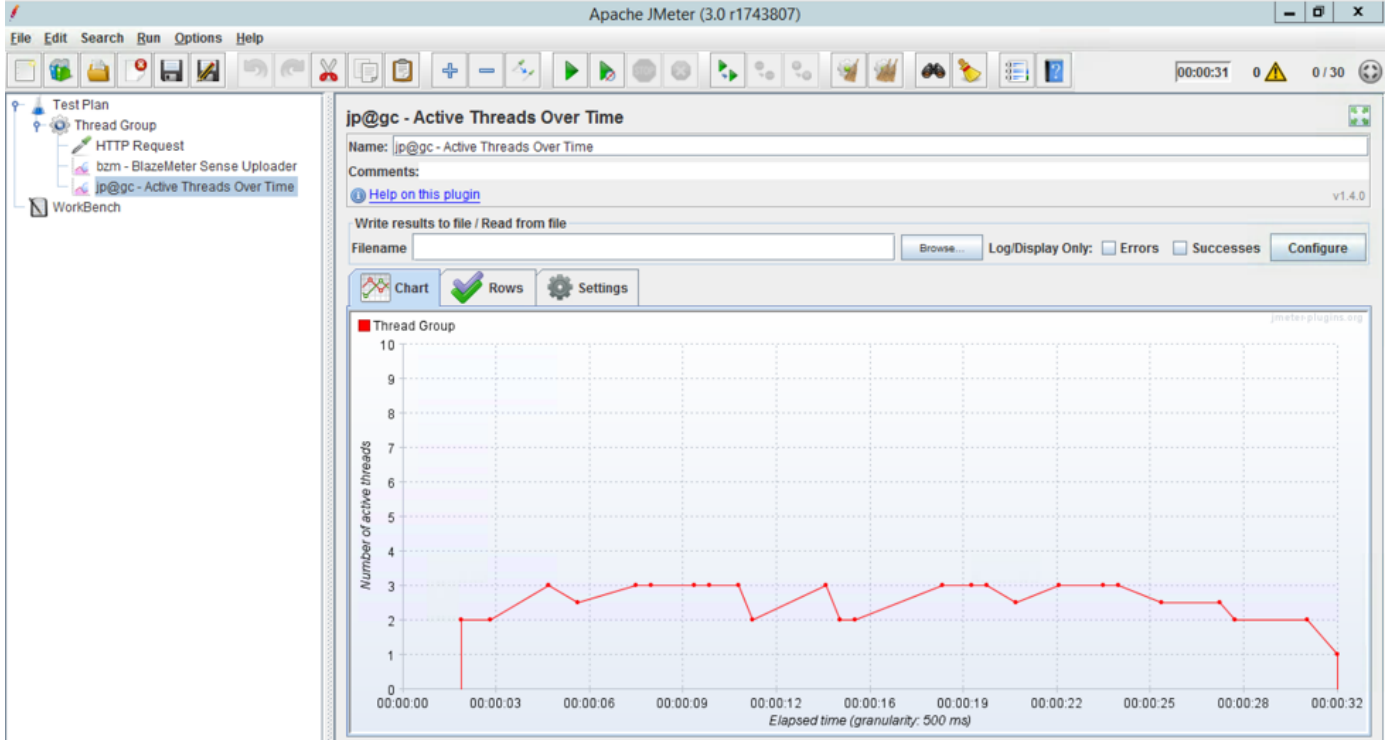
Unfortunately, JMeter doesn't provide a listener to track concurrency. So, if you would like to see users arrival plotted, you might want to consider 3rd party solutions, like:

- [Active Threads Over Time Listener](#) - the "normal" JMeter listener, which can be used standalone to visualize actual concurrency.

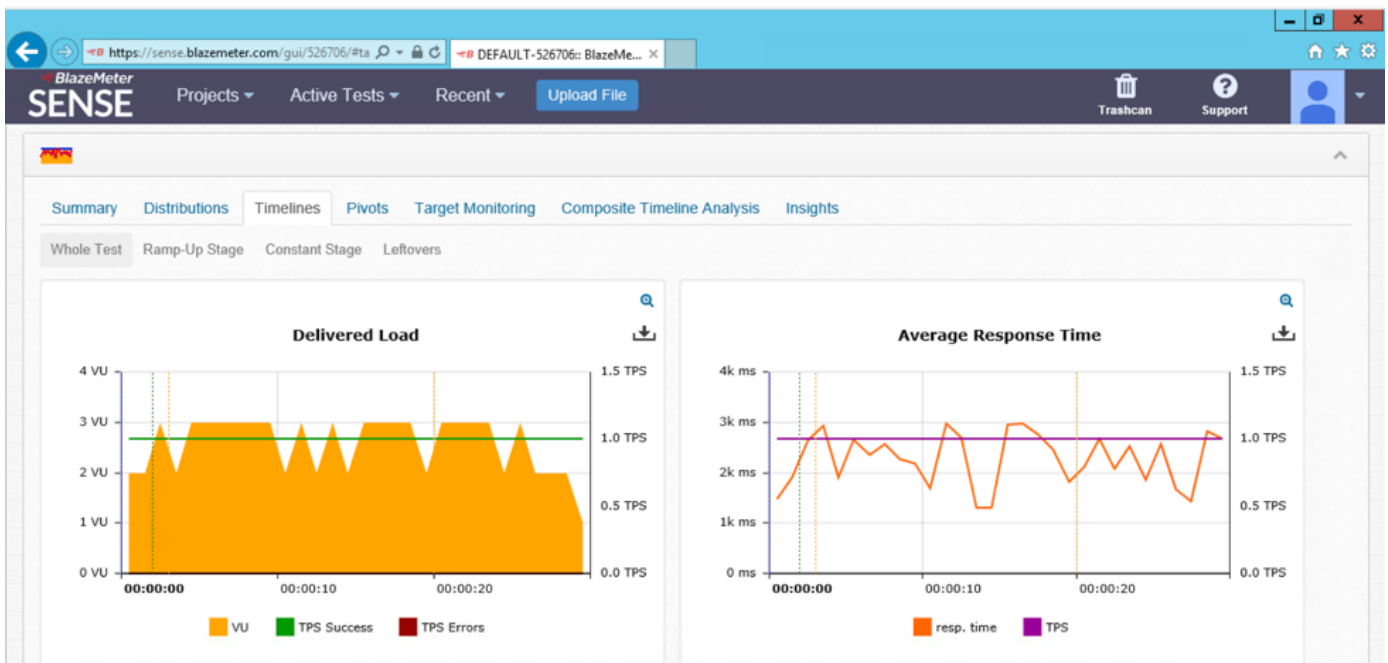
- [BM.Sense Uploader](#) - uploads test results upon completion to [BlazeMeter Sense](#) - an online JMeter test results visualization and analysis platform.

Both are available in the [Standard Set](#) of the [JMeter Plugins](#).

Let's kick off the test and see how much load we will be able to deliver, using the above settings:



If you look into the BlazeMeter Sense dashboard, you will see the same figures on the "Delivered Load" chart:



As can be seen, the actual concurrency was between 2 and 3 virtual users. Why is that?

To answer this question, we need to take a closer look at

how JMeter works:

First, a few basic terms:

● **Number of Threads** - each JMeter Thread represents a virtual user. The number of threads = the number of virtual users

● **Ramp-up Period** - the time frame (in seconds) for all requests to start. All the threads specified in the "Number of Threads" input will start within "Ramp-up period". For instance:

- 30 threads and 30 seconds ramp-up: each second JMeter will start 1 Thread, until all threads are started by the time the 30 seconds are up
- 30 threads and 15 seconds ramp-up: each second 2 Threads are started
- 30 threads and 60 seconds ramp-up: every 2 seconds, 1 Thread is started

● **Loop Count** - the number of iterations for each Thread.

JMeter acts as follows:

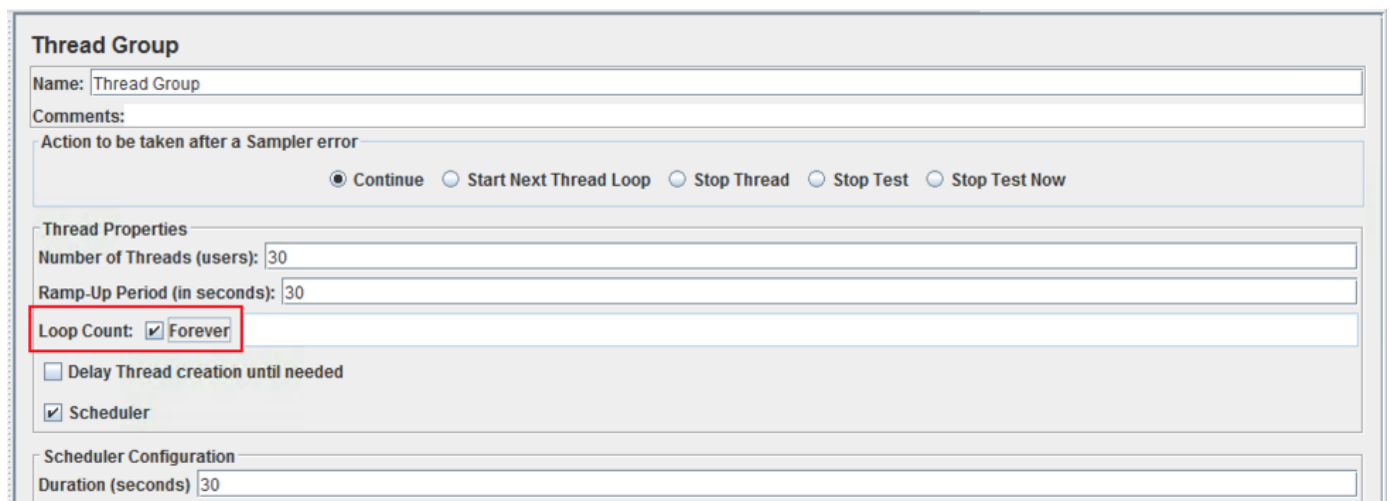
1. All Threads specified in the "Number of Threads" for the Thread Group are kicked off within the bounds of the "Ramp-up Period"

2. Each Thread starts executing [Samplers](#) upside-down (or according to the [Logic Controllers](#), if any)
3. When the last Sampler is finished, the Thread starts over (if there are any Loops left)
4. When there are no more Samplers to execute and no Loops to iterate, the Thread shuts down.

What happened here and why is the concurrency so low?

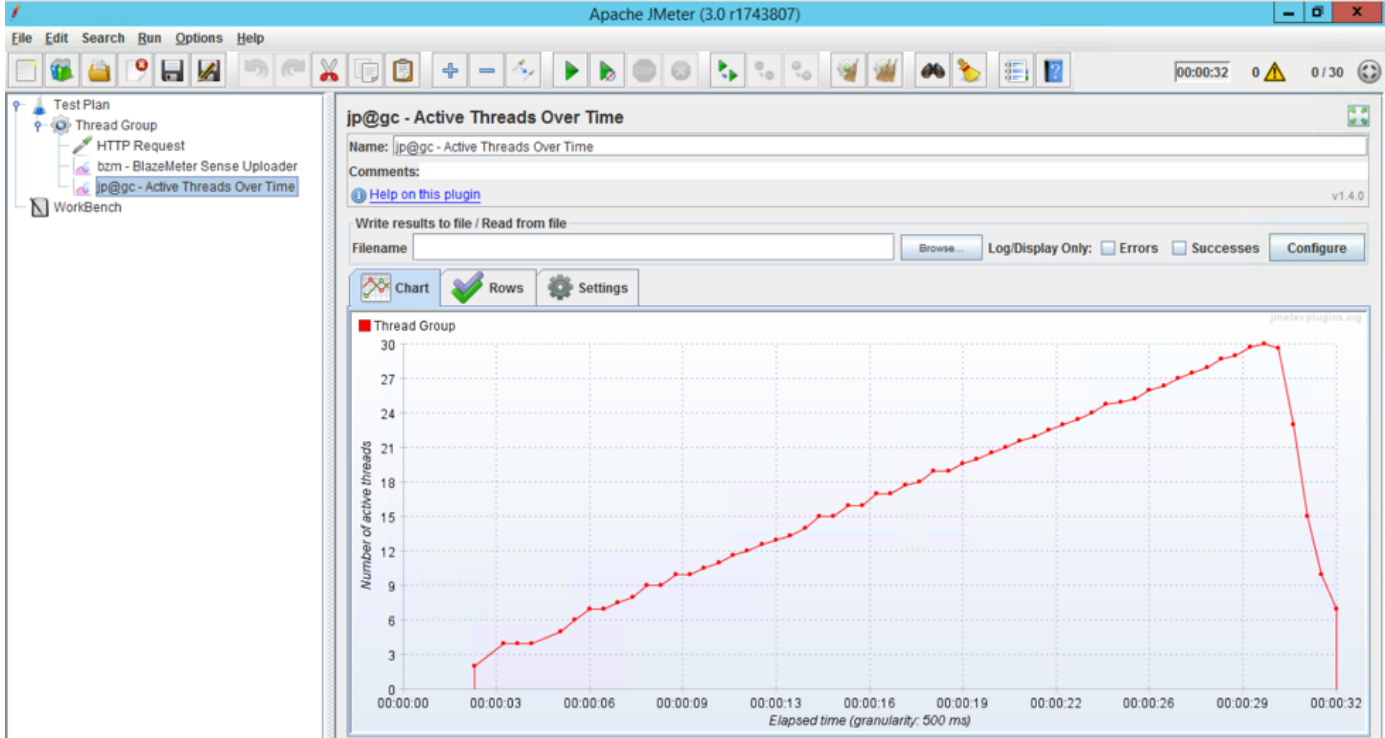
Because some threads have already finished their job and were terminated, while others haven't started yet.

To achieve the desired concurrency of 30 virtual users - provide JMeter with enough loops so the threads (virtual users) that have already executed all the samplers in scope can start over, while others are still arriving. All you need to do is to tick the "Forever" box next to "Loop Count".



The screenshot shows the 'Thread Group' configuration window in JMeter. The 'Name' field is 'Thread Group'. The 'Comments' field is empty. The 'Action to be taken after a Sampler error' section has radio buttons for 'Continue' (selected), 'Start Next Thread Loop', 'Stop Thread', 'Stop Test', and 'Stop Test Now'. The 'Thread Properties' section has 'Number of Threads (users): 30' and 'Ramp-Up Period (in seconds): 30'. The 'Loop Count' section has a checkbox for 'Forever' which is checked and highlighted with a red box. Below it are checkboxes for 'Delay Thread creation until needed' (unchecked) and 'Scheduler' (checked). The 'Scheduler Configuration' section has 'Duration (seconds): 30'.

Let's see what the test execution looks like now:



We can see that when there are enough loops provided, each newly onboarded thread is added to the others which are already working. We recommend you keep that in mind while designing your load test scenario.

Final 2 Cents

1. If you're providing the "Loop Count" value dynamically, i.e. via the [JMeter Property](#), the numeric equivalent of "Forever" will be -1.

2. The easiest way to install aforementioned Active Threads Over Time and BM.Sense Uploader listeners (as well as any other JMeter Plugins) is using the [JMeter Plugins Manager](#)

Let us know if you have any questions or comments, in the comments box below.

